Benefits of Operational Consideration into the Guidance, Navigation, & Control Design of Spacecraft

Mark C. Morris*
United Space Alliance, LLC, Houston, TX 77058

Greg N. Holt, Ph.D.†
NASA Johnson Space Center, Houston, TX 77058

The following paper points out historical examples where operational consideration into the GN&C design could have helped avoid operational complexity, reduce costs, ensure the ability for a GN&C system to be able to adapt to failures, and in some cases might have helped save mission objectives. A costly repeat of mistakes could befall a program if previous operational lessons, especially from operators of vehicles with similar GN&C systems, are not considered during the GN&C design phase of spacecraft. The information gained from operational consideration during the design can lead to improvements of the design, allow less ground support during operations, and prevent repetition of previous mistakes. However, this benefit can only occur if spacecraft operators adequately capture lessons learned that would improve future designs for operations and those who are designing spacecraft incorporate inputs from those that have previously operated similar GN&C systems.

I. Introduction

The application of historical "lessons learned" becomes crucial when designing the Guidance, Navigation, and Control (GN&C) portion of a spacecraft, whether for human or robotic missions. Implementing previous "lessons learned" can reduce the overall risk to a program in terms of technical cost, schedule, mission success, and flight safety. Operational insight provides additional information for design trades, especially when relating to GN&C systems. Designs that include operational consideration are more adaptable to real-world environment failures.

Both spacecraft designers and spacecraft operators are critical members of a spacecraft project and each have their own strengths in executing a spacecraft concept to fruition. History has shown that a balance between the two is demanded in order to optimize the overall cost of a program. Considering operations during the system design can aid a program much as an understanding of the system design can aid during operations. History has often shown that if spacecraft operators had a better understanding of the design of a spacecraft or system being operated, the repeating of previous mistakes could be avoided. However, this paper addresses the other side of the coin, encouraging spacecraft designers as a whole to consider the operational aspects of the spacecraft while in the design process.

II. Overview of Operational Consideration

When designing any system, a key component to consider is how the system will eventually be used. Although sounding simple in theory, it can become quite difficult in practice for complex systems. For a simple system it is likely that the entire design team will be the same team that operates the system; however, for a complex system the entire team is broken up into small subgroups, each with their own specific task (e.g. Guidance, Navigation, and Control (GN&C)), and it is likely that the design teams will not operate the system. It is also easy for a design subgroup to get overly focused on how to optimize their design portion of the entire system without considering the

* Onboard Navigation Shuttle Flight Controller, Navigation & Flight Design Integration Group, Mail Code USH-485L, 600 Gemini Ave. AIAA Member.

† Ascent/Entry Navigation Integration Lead, Mission Operations/Flight Dynamics, Mail Code DM42. AIAA Member.

operational impacts and/or limitations. Unfortunately the optimum design does not always balance with what is optimum for operating the system and in the end can increase life cycle costs. Those on a design team and those on an operations team each possess a unique set of skills and experience critical to the overall success of the program. When integrated properly, the expertise from both sets of skills can help lower risk (cost, schedule, development, mission success) and in some cases save programs from repeating failed moments in spacecraft history. Additionally, upholding operational consideration during the design process of a system can bring about design flexibility and margin into a spacecraft to enable mission success in the presence of failures.

Maintaining operational consideration during the design process of a system can be accomplished in many different ways. The most simplistic is reminding the design team at different stages of the design process how the system being designed will be used and the additional systems with which the design will be interacting. Considering operations early in the design allows the design team to account for improvements before the design has matured enough and become difficult to change. Programs that generate requirements to manage the design particularly benefit from operational consideration during the requirements definition phase because it is a way to specifically account for lessons learned and best practices from previous programs. Accounting for operational consideration during the requirement definition phase ensures robust requirements and reduces operational limitations. A more advanced approach to maintaining operational consideration would be to bring in outside experts, in operating systems similar to what is being designed, to help aid the design team in accounting for undesirable operational limitations made at the design phase level and improve upon previously discovered operational limitations. This approach may add some additional cost to the initial part of the program (i.e. the design portion); however, when used and accounted for properly it will benefit the program in the long run. Spacecraft systems will have failures from time to time and the use of operational consideration will not prevent all spacecraft system failures. Even if the programs used as examples in this paper used operational consideration during design, it is likely that some of the failures would have still occurred. However, it is the intent of this paper to show the reader that if operational consideration was used for these programs a percentage of these failures would have been avoided. This same principle applies for future programs.

This paper presents a subset of the available benefits of using operational consideration during the design of spacecraft. Table 1 lists a summary of historical problems with programs and operational consideration discussed in this paper.

Table 1. Historical list of problems with programs and operational consideration

Occurrence	Program	Issue	Operational Consideration
January 1994	Clementine	Computer froze, disabling the fault management software. Patches were prepared but not uploaded.	It is recommend to always have a "backdoor" capability to communicate with the spacecraft.
April 1999	Titan IV	A roll filter manually entered in the transfer vehicle's avionics database had an exponent error (effectively misplaced a decimal point) which caused control loss.	Typically for any manually entered parameter or an "on the fly" uplinked parameter a Quality Assurance (QA) process is put into place that will minimize the possibilty of user error and preapprove the acceptable value of the parameter. This could be applied for hard-coded design parameter values in the Flight Software.
December 1998	Mars Climate Orbiter (MCO)	Burned up because of unit mix-up in the navigation software.	Having someone on the design team or working with the design team who has an idea of what numbers to expect can save a program.
April 2005	Demonstration of Autonomous Rendezvous Technology (DART)	Spacecraft failed to rendezvous with its target satellite because of a combination of GPS software bug and flawed navigation design.	No matter how automated a system is intended to be, the ability to insert a "break" during an automated execution is critical for contingency purposes.
STS-79 (1996) - STS-121 (2006)	Space Shuttle GPS Implementation	GPS was added as a Position, Velocity, and Time (PVT) State replacement for the Navigation Solution with the intent to keep the design simple; however, this resulted in undesired operational complexity.	If a program considers operational complexity during the design phase, finding a balance between optimizing the design and operation can benefit a program greatly in the long run.

III. Ensuring Command Capability and Using History to an Advantage

The computer onboard the Clementine spacecraft (officially called the Deep Space Program Science Experiment (DSPSE)) froze immediately after a thruster was commanded to fire. After leaving lunar orbit, a "watchdog" algorithm designed to stop the thrusters from excessive firing could not execute and Clementine's fuel ran out on May 7 at 14:39 UTC (9:39 AM EST). This left the spacecraft spinning at about 80 RPM with no spin control. The preliminary analyses of the returned lunar data suggest that valuable scientific measurements were made on several

important topics but that the Committee on Planetary and Lunar Exploration's (COMPLEX) highest-priority objectives for lunar science through Clementine were not achieved.[1-3]

Among many other lessons from this example, command uplink capability is an important part of being able to operate any vehicle from the ground. From an operational standpoint it is vital to always reserve this capability, when possible, despite what potential failures may occur. This lesson has proved beneficial to a variety of space vehicles. An example would be the Mars Rovers Opportunity and Spirit, which despite their ability to navigate from point A to point B and execute an entire sol (Martian day) worth of activities without user interaction, found the ability to receive commands from the ground useful during dust storms and when limit power supply was able to check life signs and make adjustment to planned operations as needed on the fly.[4,5] Had the Clementine design team been advised to reserve the capability to send commands (e.g. reboot the computer) to the vehicle whenever possible despite a computer freeze or other potential failures the spacecraft might have been able to continue completing all of its main objectives.

A future program, Near Earth Asteroid Rendezvous (NEAR), learned a key lesson based on the Clementine incident: the watchdog function should be hard-wired in case of a computer shutdown. As it happened, NEAR suffered a similar computer crash during which its thrusters fired thousands of times, but each firing was instantly cut off by the still operative watchdog timer.[1] The NEAR mission was deemed a success. If the NEAR program was not aware of the Clementine malfunction it could have also suffered the same fate. As this application from the same example illustrates, insights from past lessons experienced during operations, whether attributed to design or operational decisions, are of considerable value to future designs. Information from failures (and near failures) can influence important design decisions and prevent the same mistakes from being made over and over.[1]

IV. Quality Assurance & Expected Value Range

On April 30,1999 a Titan IVB was launched; however, the Titan Centaur upper stage (TC-14) began to tumble uncontrollably during the second main engine burn. The Reaction Control System (RCS) depleted the vehicle propellant during the transfer orbit coast phase attempting to compensate for the attitude errors. The third engine burn terminated early due to the tumbling vehicle motion. As a result of the anomalous events, the Milstar satellite was placed in an undesired low elliptical final orbit, as opposed to the intended geosynchronous orbit. After several days of satellite life saving effort by Air Force and satellite contractor personnel at Schriever Air Force Base, Colorado, the Milstar satellite was declared a complete loss by the acting Secretary of the Air Force on 4 May 1999. The cause of the failure was determined to be an incorrect roll rate filter constant in the control-system software that caused radical roll errors. The erroneous constant resulted from a misplaced decimal point in the upper stage software.[6]

When a given parameter is selected to be used onboard, especially for something to be uplinked “on the fly”, a rigid Quality Assurance (QA) process is common practice during operations or in the events leading up to operations (e.g. initialization loads uplinked onboard prior to launch). This process minimizes the possibility of user error and preapproves the acceptable value of the parameter. Although it is important not to make the QA ineffective due to the diffusion of responsibility, this could be applied for hard-coded design parameter values in the Flight Software. This would apply an operational concept to benefit the design and ensure its robustness.

This example also reemphasizes the significance of integrated testing and introduces the importance of having support on the design teams that have a general idea of what values to expect in the testing results and how the vehicle should generally behave in a given environment. This is not to say that all design teams take the testing results for what they are without truly analyzing the results to see if they make sense. The implication is that those that purely work on designing a system may not have the knowledge of what to expect from the system being tested. It is not uncommon during operations to have a small subset of the design team available to talk to incase unforeseen problems arrive that the designers would be more qualified to address and troubleshoot. In the same manner would not a small subset of an operations team, for a similar vehicle to the vehicle being developed, be able to provide expertise to help avoid unintentional design decisions that may increase the cost of operating the vehicle?

Another example to further emphasize the benefit of having a QA process and those on the design team with a knowledge for what to expect during testing (i.e. recognize unreasonable data) would be the Mars Climate Orbiter (MCO).

> The MCO had been on a trajectory toward Mars since its launch on December 11, 1998. All spacecraft systems had been performing nominally until an abrupt loss of mission shortly after the start of the Mars Orbit Insertion burn on September 23, 1999. On September 29, 1999, it was discovered that the small forces Delta-Velocities (DV’s) reported by the spacecraft engineers for use in orbit determination solutions was low by a factor of 4.45 (1 pound force=4.45 Newtons) because the impulse bit data contained in the Angular Momentum Desaturation (AMD) file was delivered in lb-sec instead of the specified and expected units of Newton-sec.[7]

One of the main causes for the unit inconsistency was found to be attributed to the fact that the operations navigation team was not intimately familiar with the attitude operations of the spacecraft; however, this point does not discount the fact that if someone with an operational "feel for numbers", experienced in navigation systems similar to the MCO, were involved with the design and integration testing of the design the units inconsistency might have been discovered prior to launch.

The MCO example also points out that the QA process mentioned earlier could be applied across companies in addition to within a company. Another main cause for the unit inconsistency was found to be miscommunication and incorrect assumptions made between Lockheed Martin (LM) and Joint Propulsion Laboratory (JPL).[7] If there had been a QA process in place for when data was transferred from LM to JPL that included someone who could apply a reasonability test to the numbers and/or data results perhaps the unit inconsistency could have been avoided.

V. Realistic Data & Contingency Planning

> On April 15, 2005, the Demonstration of Autonomous Rendezvous Technology (DART) spacecraft was successfully deployed from a Pegasus XL rocket which launched from the Western Test Range at Vandenberg Air Force Base, California. DART was designed to rendezvous with and perform a variety of maneuvers in close proximity to the Multiple Paths, Beyond-Line-of-Sight Communications (MUBLCOM) satellite, without assistance (autonomously) from ground personnel. During the first eight hours of the mission (through the launch, early orbit, and rendezvous phases of the mission) ground operations personnel noticed anomalies with the navigation system. During proximity operations the spacecraft began using much more propellant than expected. Approximately 11 hours into what was supposed to be a 24-hour mission, DART detected that its propellant supply was depleted, and it began a series of maneuvers for departure and retirement. Although it was not known at the time, DART had actually collided with MUBLCOM 3 minutes and 49 seconds before initiating retirement.[8]

Although the launch, early orbit, rendezvous, and departure and retirement phases were completely successful, DART failed to achieve its main mission objectives.[8]

> In DART's case, the Mishap Investigation Board (MIB) determined that the first cause for its premature retirement occurred when the estimated and measured positions differed to such a degree that the software executed a computational "reset." By design, this reset caused DART to discard its estimated position and speed and restart those estimates using measurements from the primary GPS receiver.
>
> Careful examination of the software code revealed that upon reset, the velocity measurement from the primary GPS receiver was introduced back into the software's calculations of the spacecraft's estimated position and speed. If the measured velocity had been sufficiently accurate, the calculations would have converged and resulted in correct navigational solutions. However, DART's primary GPS receiver consistently produced a measured velocity that was offset or 'biased' about 0.6 meters per second from what it should have been. This had the unfortunate effect of causing the calculations, which were being performed autonomously, to once again diverge until the difference became unacceptable to the pre-programmed computer logic. Once the limit as to how much the calculations could differ was reached, the software executed another reset. As a result, this cycle of diverging calculations followed by a software reset occurred about once every three minutes throughout the mission. These continual resets caused the incorrect navigational data that prompted excessive thruster firings and the higher than expected fuel usage.[8]

When testing a GN&C system during the design process, running simulations with "perfect incoming data" to work out the initial kinks in the system is understandable; but, it is crucial as the system becomes more mature to test using more realistic, noisy, and at times 3-6 sigma data to adequately test the robustness of the system. This also can allow proper tuning of the system, which in this case could have identified what DART would do after missing its required six meter waypoint sphere. Experts who have seen the noisiness of the data during operations can help provide insight into expectations of the data in the real environment and the consequences for assuming "perfect data". Another important observation is that the entire system should be tested for proper integration testing because this is how the system will operate. With DART it was the GPS error causing computer resets that eventually resulted in depleting the vehicle propellant. Even if the GPS velocity bias was known prior to flight, it is not expected of the GPS software designer to realize that the GPS bias would result in depleting all of the remaining onboard propellant. Only fully integrated tests with realistic data could find such problems. Operational personnel deal with these integrated problems and when this knowledge is passed down to the design teams of future programs they can benefit greatly.

Lastly with this example it is worth noting that an operational mind set can explore, "What if this goes wrong?" or "What if I want it to stop what the spacecraft is doing so I can evaluate an anomaly?" Although this thought process should be limited to realistic failures, one common implementation for protecting "when things go wrong" while incorporating a parameter into the onboard is to have auto, inhibit, and force type capabilities in the software to control. The same ability can be applied to any planned operation that is to be completely autonomous. If any type of contingency scenario presents itself during the autonomous operations it is desired to have the ability to take

the autonomous operation to "inhibit", or the equivalent. Although autonomy is a highly desired feature in the current time, one should realize the tradeoff between a small upfront cost to have an "inhibit" type capability, possibly saving the mission objectives, versus proving the complete hands off type operation and possibly fail the entire mission. As this example outlines, operational considerations during the design can help identify possible contingency situations and solutions. The earlier in the design phase that these considerations are given consideration can save a program from eating the cost later.

VI. Optimizing Design Can Come at a Cost

GPS was introduced to the Space Shuttle in the mid 1990's as part of the Department of Defense (DoD) plan to phase out TACtical Airborn Navigation (TACAN) system. Since the Space Shuttle Kalman filter was already certified, GPS was integrated into the shuttle avionics system as a separate navigation system. The design decision was to use a selected GPS state vector as a periodic replacement of the shuttle navigation state vector.[9]

> This avoided filtering GPS measurements or a position vector. There was a strong desire to avoid modification and re-certification of the flight proven entry navigation algorithms and Kalman filters in the PASS and BFS software. The state replacement architecture also permitted flying the GPS receiver in a test mode, without committing to use GPS for navigation.[9]

The decision to prevent retesting and recertifying the Kalman filter already in place onboard the Space Shuttle was an optimum decision for the design team. However, in order to ensure the GPS state vector was acceptable to replace the navigation state vector during operations required more work than the design team anticipated. The equivalent of fault detection, isolation, and recovery (FDIR) and redundancy management (RM) were redeveloped using four quality assessments (QA). The first QA provided limitations on acceptable Figure of Merits (FOM). Much analysis was required in order to determine what FOMs were acceptable for different phases of flight (i.e. ascent, orbit, and entry). The second QA provided acceptable limits on the difference between the onboard navigation state vector and the selected GPS state vector. A ramping limit algorithm was selected based on the amount of time since the last state update, whether ground or GPS.[9]

The amount of research and discussions needed to conclude on the current limits used on the Space Shuttle was large enough for this QA that it should not be overlooked. The third QA is a compare between the current and previous GPS states for a given receiver and the fourth QA, when more than one is available, is a compare of current state vectors between receivers.[9] The Shuttle community recognizes that these four QAs are not all encompassing and are not as straight forward as determining the traditional RM limit for given measurements; but, this method was determined necessary because of the decision to use GPS as a state vector replacement external of the Kalman filter. Using the GPS state vector to replace the shuttle navigation state vector was executed successfully; however, this decision called for more ground support than anticipated and the additional ground software for this support.

Although some of the cause for the implementation of these QA's could be attributed to the specific GPS receiver that was selected for the Space Shuttle, the principle is still valid. A program must be careful to find a balance between optimizing that design, in order to cut down on development and testing costs, and optimizing the operation of the vehicle. If the entire trade is not considered at the time the design direction is decided, a program could arguably be paving a plateau only to climb an uphill mountain later.

VII. Conclusion

In each of the previously described scenarios, there is an aspect of operational insight that may have benefited the design of each spacecraft. For Clementine it was the ability to have a "back door" capability implemented during the design phase in order to communicate to the spacecraft. For the Titan IV and Mars Climate Orbiter operational consideration could have helped to develop a quality assurance process to verify and validate critical parameters, both within a company and across companies. Additionally for these programs, having personnel on or working with the design team that had the knowledge for an expected range of values (i.e. "feel for numbers") during testing could have benefited the designs. For DART having personnel on the design team that could help test with real-environment data and implementing a rather simple protection for contingencies could have benefited the design. For the Shuttle considering the operational complexities and ground support needed for an optimum design decision could have benefited the design of implementing GPS.

Incorporating operational input in the GN&C design of spacecraft increases the ability for a GN&C system to be able to adapt to failures and reduces the overall risk to a program in terms of technical cost, schedule, mission success, and flight safety. Operational input allows a GN&C design team to take into account life cycle impacts (e.g. costs) and enables informed decisions using historical data, best practices, and lessons learned.

References

[1]Cheng, Paul and Smith, Patrick, "Learning from Other People's Mistakes," *Crosslink*, Fall 2007, pp.20, 24.

[2]Williams, David R., Ph.D., "Clementine Project Information," *NASA Goddard Space Flight Center*, 13 April 2005, URL: http://nssdc.gsfc.nasa.gov/planetary/clementine.html [cited 06 December 2010].

[3]Commitee on Planetary and Lunar Exploration, Space Studies Board, Commission on Physical Sciences, Mathematics, and Applications, National Research Council, *Lessons Learned from the Clementine Mission*, Published by National Academy Press, Washington, D.C., 1997.

[4]Norris, Jeffrey S., Powell, Mark W., Vona, Marsette A., Backes, Paul G. and Wick, Justin V., "Mars Exploration Rover Operations with the Science Activity Planner" 2005 *IEEE International Conference on Robotics and Automation* Barcelona, Spain, April 2005.

[5]Rayl, A. J. S., "Mars Exploration Rovers Update: Spirit Remains Silent, Opportunity Pushes on the Endeavour," *The Planetary Society*, 31 Oct. 2010, URL: http://www.planetary.org/news/2010/1031_Mars_Exploration_Rover_Update_Spirit.html [cited 06 December 2010].

[6]Pavlovich, Colonel J. Gregory, "Milstar Accident Investigation Board Report," *United States Air Force*, 22 July 1999, URL: http://sunnyday.mit.edu/accidents/titan_1999_rpt.doc [cited 03 December 2010].

[7]Stephenson, Arthur G., "Mars Climate Orbiter Mishap Investigation Board," *George C. Marshall Space Flight Center*, URL: ftp://ftp.hq.nasa.gov/pub/pao/reports/1999/MCO_report.pdf [cited 09 December 2010].

[8]Marshall Space Flight Center, "Overview of the DART Mishap Investigation Results," *Summary of DART Accident Report* [Press Release], URL: http://www.nasa.gov/pdf/148072main_DART_mishap_overview.pdf [cited 03 December 2010].

[9]Goodman, John L., "Operational Use of GPS Navigation for Space Shuttle Entry" *IEEE/ION Position Location And Navigation Symposium (PLANS)* Monterey, CA, 5-8 May 2008.

[10]Kachmar, P. M., Chu, W., Neirinckx, P., and Montez, M., "U.S. Space Shuttle Integrated GPS Navigation Capability," *Proceedings of the Institute of Navigation GPS 93 Conference*, Institute of Navigation, Fairfax, VA, 1993, pp. 313-326.

[11]Jordan, Steven M., "Healthy Tension – The Role of Operations in the Orion Spacecraft Design" *AIAA SPACE 2009 Conference & Exposition* Pasadena, CA, 14-17 Sept. 2009.